Towards mobile spatial audio for distributed musical systems and multi-user virtual environments

Jeremy R. Cooperstock
McGill University
Centre for Intelligent Machines
3480 University Street
Montreal, Quebec
jer@cim.mcgill.ca

Mike Wozniewski
La Société Des Arts Technologiques
1195 Saint-Laurent boulevard
Montreal, Quebec
mike@mikewoz.com

Zack Settel
La Société Des Arts Technologiques
1195 Saint-Laurent boulevard
Montreal, Quebec
zack@sat.qc.ca

ABSTRACT

We developed a paradigm for interaction with spatial audio in virtual environments, where users are immersed in a three-dimensional sound-processing world. Audio signals may be steered along spatial pathways, and signal processing takes place at specific 3-D locations. Although a variety of new applications can be conceived with this framework, user mobility is typically limited because the current implementation is based on a centralized architecture in order to support the processing and rendering demands. This paper describes some of the core issues involved in the challenge of migrating this spatial audio architecture to an environment of wireless, mobile devices, so as to support multi-user interaction within distributed sonic spaces.

1. INTRODUCTION

The AudioScape project [4] allows for multiple users to be immersed in a shared virtual world. Each user has a subjective view into the environment, which is rendered in real-time according to their hardware configuration. Some participants may use an individual kiosk-style setup, with one screen for the visual display and a binaural headset for the auditory display. Others may be situated in a surround projection environment, with several loudspeakers providing spatialized audio. Regardless of the technology, one main goal is to support navigation about the virtual world and interaction with other users and various virtual objects in the scene. Unlike most virtual reality simulations, the content of our virtual world is highly musical and artistic. Typical objects include sound generators or processors such as delay boxes, flangers, harmonizers, and sound loops. In addition to the audio and visual rendering that is provided to each participant, individuals can generate their own sound signals as input to the environment, either through microphones or sound production equipment.

In previous work [6, 7, 8], we developed a framework for managing audio in 3-D space. We allow for the placement of signal processing nodes at specific 3-D coordinates, and then allow for the tuning of physical modelling so that sound can be steered precisely between them. Our next challenge is to free the user from a fixed geographic location, and allow for distributed mobile sound installations.

2. THE AUDIOSCAPE ENGINE

At the outset of this project, we investigated various APIs and toolkits for managing audio in virtual environments (e.g. OpenAL, DirectX, X3D), but found their focus limited to single-user simulation applications and simple spatialization of external audio streams (sound files, line-in, etc.). This limitation motivated the development of AudioScape, a 3-D audio architecture based on nodes that behave simultaneously as *sound sources*, which emit audio into the scene, and *sound sinks*, which collect audio at a particular location. The nodes can therefore be used as spatially located signal processing units that take an input sound signal, modify the sound, and emit the result back into the scene.

AudioScape uses PureData (Pd) [2] to manage all signal processing and interaction with audio hardware. Several external[1] libraries for Pd have been developed to manage the 3-D arrangement of nodes and produce a visual rendering of the scene. These build upon the OpenSceneGraph graphics library, which provides efficient data structures for managing 3-D content and operations including culling and collision detection.

The AudioTwist package [1] provides several editors and patches for AudioScape, allowing users to create complex *virtual audio scenes* that can function as musical instruments or interactive sonic applications. The scene may be distributed over a network, supporting multiple participants. A variety of input devices can be used to help users control and modify the audio scene dynamically in real-time, supporting applications such as audio synthesis, processing, and mastering.

3. MOVING TO MOBILE

Given the accessibility and decreasing cost of mobile technology, we are now able to move beyond geographically fixed installations, and instead, distribute the AudioScape over various mobile and wireless devices. This could enable effective full-duplex audio interaction between multiple users

[1]PureData provides an API to create new objects for use in its visual programming interface. These dynamically loaded libraries, written in C, are called *externals*.

in the augmented audio space, taking into account issues of latency sensitivity and positional awareness between individuals. Doing so requires a single modelled audio scene that includes all of the users as both positional sound sources and sinks, with user-specific computations run in a distributed manner. We note the architecturally simpler alternative of running multiple instances of the software on a single host and rendering the audio for each listener through headphones. This is, however, limited in scale, subject to increased latency as the working space grows, and ill-suited to supporting *interaction* between multiple users, for example, social games and ensemble performance.

The significant engineering challenge here is one of scalability. As the number of users increases, and in particular, if the users cluster together, there is the potential for an n^2 effect of audio interactions between them. Under these conditions, both computational load and system latency must be considered with regard to user experience. Latency tolerance limits are important with respect to individual activity (feedback) but also between multiple users [3]. In the few milliseconds of acceptable response time, the user's head orientation must be sensed, the relevant audio computations performed in the modeled space, and the sound signal played to that listener. The resulting audio must then be propagated to other nearby listeners, and mixed with their sound signal, based on their respective head positions and orientations, also, within a highly constrained time budget. It is therefore imperative that system load is self-monitored, so that steps can be taken to achieve graceful degradation (e.g. computational reduction by audio sub-sampling) rather than exceed critical thresholds of performance.

4. APPLICATIONS

The sonic and musical applications we consider in our mobile environment involve multiple users interacting in large augmented spaces. As an example, users could move about in the absolute scale of a real football field and discover a number of virtual audio objects that exist in a hybrid reality. Where appropriate, for example, in game play, the underlying physical spaces may be represented topographically in the virtual space. This permits the delineation (via auditory cues) of the *playing area*, regions of importance or game-specific topologies. Because audio is rendered individually, each user can be assigned unique roles, with their sound input or output processed differently. Furthermore, sound need not behave according to the true laws of physics. This allows for interesting and creative interactions to take place. For instance, it might be possible to find a virtual sink node that *teleports* sound to another location, or an effect box that splits a sound signal into a number of harmonized components that each propagate in new directions.

5. ENVISIONED ARCHITECTURE

5.1 Hardware Platform

In a spatial audio context, a set of headphones allows sound to be rendered with sufficient detail to support localization, while an attached microphone enables the addition of sound input to the scene. In order to steer sound capture for binaural spatialization and sound emission of the microphone in the appropriate directions, the 3-D position and orientation (steering direction) of the user must be maintained at all times. Position tracking technologies we are considering include GPS and Local Positioning Systems, using both RF and video, as well as motion capture systems. Since the audio must be distributed to other users, some form of wireless transceiver is also required. For more complex tasks, where visual feedback is needed for users to align themselves with sound processing entities in the scene, some form of visual display may also prove valuable. The inclusion of other controllers, such as a gamepads, to facilitate additional parameter modifications could also be useful. This suggests a minimal set of hardware requirements comprising a headset, microphone, location and orientation sensor, with wireless transceiver capability and support for serial and/or USB peripheral devices.

5.2 Processing and Latency

A significant question we are investigating is the extent of processing that should take place locally, on distributed mobile devices, versus centrally, on a single computer. Highly portable devices such as the Gumstix [5], with mid-range processing power, offer a platform that could support the hardware requirements listed above, although experimentation is needed to determine the application limits of their performance. More powerful processors, found on heavier PDAs and Pocket PCs, may be more suitable for the demands of interactive audio applications, in particular should it be necessary to maintain a copy of the entire state of the virtual world, and render the subjective output for each user, locally. These devices can also include a display and a powerful operating system, allowing for a potentially richer set of interactions. It is worth noting that PureData has a port called PDa, which runs on many PDAs by converting all operations to fixed point arithmetic. At the other end of the spectrum, wireless receivers, using RF or Bluetooth, could be deployed to receive individual audio signals for each user, computed on a powerful central server or cluster, with no local processing requirements. While video is more bandwidth-intensive, similar hardware options exist, such as those based on surveillance technology.

An important issue that must be addressed in such architectures is that of latency. Mobile applications often incur significantly greater latency than is tolerable for musical interaction. Typical compression algorithms are well suited for bandwidth reduction but unusable in an interactive audio environment. Instead, raw audio and any necessary sensor information, such as orientation information, which modulates the resulting sound, must be communicated between distributed users as quickly as possible. While further investigation is clearly in order, an initial assessment of existing wireless audio technologies suggests that there are few options currently available that satisfy these requirements.

One such possibility is the transmission of uncompressed PCM over IEEE 802.11, which minimizes latency by avoiding compression algorithms, and would thus be suitable to support true musical interaction between multiple participants in a shared sonic environment. Because of the relative low cost and widespread availability of the technology, this is likely to be a useful approach, at least for prototyping purposes. However, WiFi draws significant power, implying the need for larger, heavy batteries on mobile devices. An interesting alternative is offered by Fraunhofer's Ultra Low Delay Audio Coder, which would significantly reduce bandwidth requirements and thus, permit the use of less

power-intensive wireless protocols.

The architectural choice depends on many factors, such as the processing and power requirements, latency tolerance, number of users, and cost of deployment. Some of these issues will be addressed in the following sections.

5.2.1 Centralized approach

The spatial audio system we envision could easily be deployed as a centralized architecture. The entire description of the scene would be maintained on one machine, which accepts all audio signals and sensor information from devices on the network. This server continuously maintains and updates the state of the world, and sends appropriate content to passive renderers for their output. This simple approach can support a modest number of simultaneous users, perhaps dozens with commodity hardware. It offers the advantage of utilizing low-cost, passive rendering hardware, obviating the need for expensive and potentially delicate PDAs with the alternative of simple, low-cost receivers. This model suggests the use of wireless microphones for audio acquisition and external sensing systems for position and orientation information, for example, video-based capture from several overhead cameras.

However, as noted above, this solution is not extensible to larger scale installations or performances where hundreds of users need to be supported. Further drawbacks include the lack of precision attainable for tracking position and orientation of participants, with the potential result of a disappointing user experience. Errors in head orientation, in particular, are likely to be disruptive to the localization of sound sources in the environment. Another challenge develops as the number of users increases, since this entails a corresponding increase in the number of dedicated transmission channels, and in turn, potential concerns of signal interference.

5.2.2 Distributed rendering

The distributed alternative entails maintaining a partial or full description of virtual world state on individual mobile computing devices associated with each user. Attached sensors would transmit (ideally via multicast or broadcast) their information to peer devices, providing world updates to all renderers as relevant. Unfortunately, even with multicast grouping capability, it is difficult to determine a priori whether any particular update is relevant to each peer. In our framework, spatial proximity is not necessarily an indicator of relevancy, since sound can disobey the laws of physics and travel unlimited distances without any decay. Furthermore, as connections between sound sources and sinks are defined explicitly by the user, sinks are not always connected to nearby sources. Thus, sending orientation information between these nodes would be superfluous. As a result, mobile devices with limited computational resources could be overwhelmed by updates that are irrelevant for their *view* of the world. To address this problem, either a low-cost filtering mechanism that can quickly determine the relevance of any given update, or a higher-level controller that determines what information should be conveyed to each renderer is required.

5.2.3 Hybrid approach

The ideal solution, at least based on current technology, is likely to be a hybrid approach, in which a centralized server maintains world state but only computes a high-level representation of the audio environment rather than the actual sound output. For example, the server could broadcast all raw sound streams and selectively transmit to each user a description of all proximal sound nodes that have an observable effect for the specific receiver. Local computation, performed at the receiver, would then read from a connected orientation sensor and create the appropriate filters to simulate the spatial effects that should be applied to each raw signal based on the higher-level scene description. This offers the advantages of reducing computation at the end points while minimizing latency between sensory input and audio output.

5.3 Power Requirements

Along with issues of latency, the power requirements of mobile and wireless technology must be considered. In general, power consumption increases with bandwidth capability. Ultra-wideband (UWB) transmitters for instance, are an attractive option in terms of their transmission capability, which could easily support multiple audio and video streams. One could attach a USB headset, webcam, and joystick to a single, compact UWB hub, to obtain a full architectural solution following the centralized rendering approach described above. However, currently available devices are power-hungry[2] making them generally unsuitable for mobile, wireless applications. In the more distributed rendering approaches, we find that as processing power increases, so does power consumption. Again, the hybrid approach likely offers a suitable balance between power consumption and versatility, although further exploration in this area is clearly required.

6. DISCUSSION

We have described the preliminary steps to deploy our AudioScape engine in a mobile multi-user setting. There are several choices to be made, both for present development purposes and in the future, as the constituent technologies evolve. The goal of providing a subjective rendering of a spatialized virtual audio scene to multiple users is a challenge involving efficient distribution of the scene description along with localized processing. Fortunately, support already exists for appropriate audio rendering software, notably, PureData, on mobile computational platforms with wireless transmission and reception capability, as well as integration of peripheral sensors. Design choices include alternative transmission systems, distribution of computational load, tradeoffs between bandwidth, power requirements, and system latency, each of which is likely to be dependent, at least in part, on requirements of the specific set of applications for which the system is targeted.

7. REFERENCES

[1] audiotwist website (not yet live). http://tot.sat.qc.ca/.

[2] Puredata. www.puredata.info.

[3] C. Chafe, M. Gurevich, G. Leslie, and S. Tyan. Effect of time delay on ensemble accuracy. In *Proceedings of the International Symposium on Musical Acoustics*, 2004.

[4] J. Cooperstock. Audioscape. http://www.cim.mcgill.ca/sre/projects/audioscape/.

[2]For example, the Belkin UWB USB hub consumes 5 Volts at 2.5 Amps.

[5] Gumstix. Gumstix. http://www.gumstix.com/.

[6] M. Wozniewski, Z. Settel, and J. R. Cooperstock. A framework for immersive spatial audio performance. In *New Interfaces for Musical Expression (NIME), Paris*, pages 144–149, 2006.

[7] M. Wozniewski, Z. Settel, and J. R. Cooperstock. A paradigm for physical interaction with sound in 3-D audio space. In *International Computer Music Conference*, 2006.

[8] M. Wozniewski, Z. Settel, and J. R. Cooperstock. User-specific audio rendering and steerable sound for distributed virtual environments. In *Proceedings of International conference on auditory displays (ICAD 2007)*, 2007.